**Một số lệnh git cơ bản**

Những lệnh cơ bản nhất:

**$ git add –all** *//add tất cả*

**$ git commit –m”<Message>”** *//commit lên theo dõi*

**$ git commit –a –m”<Message>”** *// gộp add and commit*

**$ git push <bí danh> <branche\_name>** *// push lên*

**$ git clone <link>** *//tạo clone*

Lệnh biến 1 folder thành 1 kho lưu trữ local: **$ git init**

Nếu bạn muốn tự khởi tạo Git của riêng mình và muốn nó liên kết với GitHub, thì bạn phải tự tạo trên GitHub trước, copy URL của repo, sử dụng lệnh **git remote add origin <URL>,**thay URL của repo vào “<URL>”. Từ đó, bạn có thể thêm, commit và push đến remote repo của bạn.

git remote -v

git remote add origin <url>

git remote set-url origin <url>

Link: <https://github.com/rajahes/CodeBase.git>

**Làm việc với Branch**

Để tạo mới 1 branch:

**$ git branch <tên branch>**

Có thể liệt kê danh sách branch:

**$ git branch**

Có nhiểu branch , giờ muốn chuyển qua branch khác để code mà không ảnh hưởng đến code hiện tại (trên branch Master) thì dùng lệnh này để chuyển sang branch mới :

**$ git checkout <tên branch cần chuyển qua>**

Nếu chưa có branch cần chuyển thì có thể vừa tạo vừa chuyển:

**$ git checkout –b <tên branch cần chuyển qua>**

Sau 1 thời gian cập nhật vào branch mới(ex: mad) , bây giờ bạn muốn ghép (merge) code lại vào cây gốc(master). Trước tiên bạn phải checkout ra khỏi branch hiện tại để vào branch master , sau đó dùng lệnh merge để ghép vào master:

**$ git checkout master**

**$ git merge mad**

Để kéo phần update mới về :

**$ git pull // git pull origin <branch>**

Để xem lại lịch sử những lần commit :

**$ git log**

Ngoài ra , mỗi lần commit sẽ sinh ra 1 đoạn mã (hash) random ví dụ như: 56c746bcv56c748h547744747

Đoạn hash này coi như là Id của lần commit đó , nó cho biết thông tin lần commit đó. Để xem những thay đổi đó ta dùng lệnh:

**$ git show < đoạn hash>**

Nếu một ngày bạn commit code và push lên remove repository , và làm hỏng ứng dụng . Thay vì sửa lại dưới local và cứ để lỗi ở trên remote , sau khi sửa xong thì push cái mới lên , bạn lại muốn xóa bỏ dữ liệu của lần commit trên remote đó . Cách đơn giản nhất là bạn đưa HEAD của source về lại lần commit trước (trước commit muốn xóa), sau đó push code của lần commit trước lên , sau đó đưa HEAD quay về lần commit muốn xóa.

**$ git reset –hard HEAD@{1}**

**$ git push –f origin master**

**$ git reset –hardHEAD@{1}**

Ví dụ bạn có 4 commit trong máy local của bạn (chưa push trên GitHub). Các commit của bạn trông thật vũng về và thiếu quyết đoán. Bạn có thể dùng lệnh **rebase**để gộp tất cả các commit lại với nhau thành một commit ngắn gọn, vắn tắt hơn.

git rebase -i HEAD~4

Câu lệnh trên sẽ mở editor mặc định ở máy tính của bạn (thường là Vim), với nhiều cách lựa chọn cách bạn muốn thay đổi commit. Nó sẽ nhìn giống như phía dưới:

pick 130deo9 oldest commit message

pick 4209fei second oldest commit message

pick 4390gne third oldest commit message

pick bmo0dne newest commit message

Để gộp tất cả các commit đó lại, chúng ta cần sửa “pick” thành “fixup” để hợp các commit và loại bỏ các commit messege. Chú ý là trong vim, bạn cần phải nhấn phím “a” hoặc “i” để có thể chỉnh sửa được văn bản, để save và thoát, bạn cần nhấn phím **Esc** rồi nhấn tổ hợp phím “shift + z + z”

pick 130deo9 oldest commit message

fixup 4209fei second oldest commit message

fixup 4390gne third oldest commit message

fixup bmo0dne newest commit message

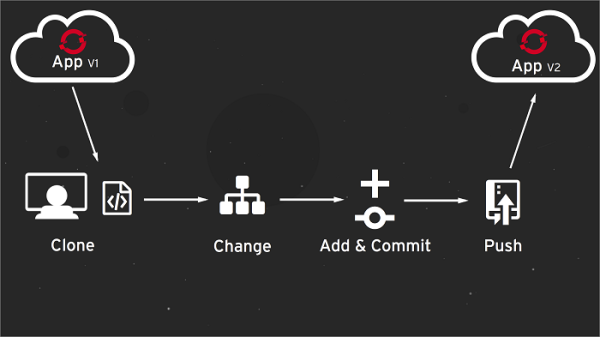
Câu lệnh này sẽ hợp nhất tất cả các commit của bạn thành một commit với messege là “oldest commit message”.

Để thay đổi commit messege, bạn sử dụng **amend**

git commit –amend

**Lưu ý**: bạn nên tránh rebase và nhét commit của bạn nếu bạn đang cộng tác trong một dự án, và có code đẩy lên GitHub. Nếu bạn bắt đầu thay đổi lịch sử các phiên bản mà không báo cho mọi người, bạn có thể sẽ làm cho cuộc sống của tất cả mọi người thành ác mộng với các lỗi rất khó theo dõi.

Quy trình làm việc 1 git:



*Quy trình làm việc của Git*